Tiago Dakuzaku

Estudo sobre o padrão Machine Readable Travel Document e o Registro de Identidade Civil brasileiro

Florianópolis - SC

2013

Tiago Dakuzaku

Estudo sobre o padrão Machine Readable Travel Document e o Registro de Identidade Civil brasileiro

Trabalho de Conclusão de Curso apresentado como parte dos requisitos necessários para a obtenção do grau de Bacharel em Sistemas de Informação da Universidade Federal de Santa Catarina.

Universidade Federal de Santa Catarina – UFSC

Departamento de Informática e Estatística

Bacharelado em Sistemas de Informação

Orientador: Jean Everson Martina

Coorientador: Ricardo Felipe Custódio

Florianópolis - SC

2013

Tiago Dakuzaku

Estudo sobre o padrão Machine Readable Travel Document e o Registro de Identidade Civil brasileiro

Trabalho de Conclusão de Curso apresentado como parte dos requisitos necessários para a obtenção do grau de Bacharel em Sistemas de Informação da Universidade Federal de Santa Catarina.

Prof. Leandro José Komosinski, Dr.
Coordenador do curso

Eduardo dos Santos
Banca examinadora

Rick Lopes de Souza
Banca examinadora

Florianópolis - SC
2013

Resumo

Um passaporte eletrônico (ou “e-passport“) é uma combinação entre um documento físico e um eletrônico, possuindo várias informações do usuário. Os dados (nome, assinatura, foto, digitais, etc) podem ser utilizados para a autenticar a identidade do usuário sem a necessidade de consultas a sistemas externos. Este modelo de passaporte tem sido utilizado em vários países, implementado com base nos padrões Machine Readable Travel Document (MRTD) especificados pela International Civil Aviation (ICAO).

No Brasil, este modelo de identificação será implementado pelo Novo Registro de Identidade Civil (RIC). Além de identificar o número do RIC, esse novo cartão também irá reunir dados de outros documentos, como RG (Registro Geral), CPF (Cadastro de Pessoas Físicas), etc. Com a adoção do RIC, todos os estados brasileiros passarão a utilizar o mesmo sistema para emitir a nova identidade, e os dados essenciais serão mandados para uma central que vai formar o Cadastro Nacional Único.

Este estudo tem como objetivo compreender o funcionamento da autenticação e segurança dos dados contidos dentro do cartão inteligente (smartcard), assim como explorar as falhas de segurança já conhecidas desta tecnologia e desenvolver protótipos de aplicações dentro do cartão que acessem apenas dados específicos.

Palavras-chaves: Palavras-chave: ePassport, RIC, MRTD, ICAO, smartcard, autenticação digital.

Lista de ilustrações

Lista de abreviaturas e siglas

MRZ	Machine Readable Zone
PKI	Public Key Infrastructure
PKD	Public key directory
APDU	Application Protocol Data Unit
CRL	Certificate Revocation List
DES	Data encryption standard
SHA	Secure Hash Algorithm
MAC	Message Authentication Code
LDS	Logical Data Structure
DS	Document Signer
DSO	Document Security Object
CDS	Document Signer Certificate
CA	Certification Authority
CSCA	Country Signing CA
CCSCA	Country Signing CA Certificate

Sumário

Introdução

Contextualização

Com a evolução dos dispositivos móveis, observa-se uma tendência de migração no processamento e armazenamento de dados centralizados, passando a serem cada vez mais distribuídos de forma com que a informação esteja sempre disponível. Um smartcard (cartão inteligente) pode conter uma quantidade significativa de informações que ficariam sempre acessíveis ao usuário quando lhe for requisitado, de forma simples e segura.

A crescente demanda por autenticação pelos mais diversos serviços do cotidiano (cadastros, assinaturas, etc) requerem cada vez mais a automatização dos processos e integração dos registros. Atualmente no Brasil, vários documentos são atribuídos à mesma pessoa e a validação de cada registro é feita de forma independente e geralmente manual.

Portanto, o uso de um smartcard como documento pessoal facilitaria não só o acesso aos dados básicos e autenticação do titular, mas também eliminaria processos lentos e manuais. Outros potenciais usos para este documento eletrônico seriam aplicações que poderiam ser gravadas no mesmo cartão (encapsuladas), complementando a autenticação (ex.: controle de presença, histórico médico, fechaduras eletrônicas, etc).

Apesar dos evidentes benefícios de um smartcard como documento, é de extrema importância garantir a segurança dos dados contidos no cartão, além de garantir sua autenticidade. É contestado em vários países quais informações devem estar contidas em um passaporte neste novo modelo e como isso irá impactar no direito de privacidade.

Se os dados não estiverem criptografados e se não houverem procedimentos de controle de acesso e detecção de fraudes confiáveis, o novo documento seria muito menos confiável do que o documento impresso tradicional.

Pensando nisso, a International Civil Aviation Organization (ICAO) criou o Doc 9303 contendo todas a especificações de como devem ser criados os passaportes com suporte a biometria e legíveis por máquina (Machine Readable Travel Documents, MRTD).

O modelo proposto pela ICAO foi escolhido por ser o novo padrão internacional de passaportes, com um grande crescente número de países participantes. Além disso, o Brasil adotou este modelo para seu novo documento de identidade civil (RIC), por ele atender a todas a necessidades de segurança e autenticação biométrica.

Objetivo

O objetivo principal deste trabalho consiste no estudo de um aplicativo que simule o processo de autenticação de um usuário seguindo o padrão ICAO9303. Com base no código e em estudos realizados por outras pesquisas em MRTD, serão feitas análises das vulnerabilidades e propostas de melhorias neste modelo.

Em complemento as análises sobre o modelo, também é apresentada uma aplicação que adiciona mais informações ao RIC, através de certificados de atributos. Esta nova funcionalidade permitiria ao titular vincular novos dados pessoais de forma independente do governo federal e sem compremeter a segurança do documento.

Objetivos específicos

A centralização de várias informações em um único cartão e a possibilidade de validação automatizada é de evidente praticidade e conveniência. Esta grande responsabilidade em um único cartão demanda igual confiabilidade e segurança. Este estudo tem como principal objetivo responder a questões como:

- O RIC é totalmente seguro contra fraudes?
- Como é garantido o acesso autorizado à leitura do cartão?
- Em caso de perda/roubo, como os dados serão protegidos?
- Em caso de dano ou defeito no cartão, a autenticação ainda é possível?
- Qual a viabilidade técnica real?
- Quais funcionalidades adicionais podem ser incluídas?

Justificativa

A autenticação tradicional por meio de documentos impressos se baseiam somente em características físicas do documento. Mesmo no caso de passaportes, onde um fiscal treinado pode consultar um terminal para averiguar os dados, todo o procedimento de autenticação é manual.

A proposta da ICAO é de melhorar os métodos adicionando etapas de verificação digital dos dados, através de assinatura digital e validação online. Entretanto, falhas de código (intencionais ou não) são impossíveis de serem detectadas por um inspetor.

Por se tratar de um novo modelo de documento sendo adotado mundialmente, a análise de segurança do padrão proposto e suas vulnerabilidades é de extrema importância.

A conveniência em se realizar autenticação automatizada (ePassports) ou possuir diversos dados centralizados em um único documento (RIC) são apenas requisitos secundários.

1 MRTD

O trabalho da ICAO em documentos de viajem legíveis por máquina (machine readable travel documents, MRTD) começaram em 1968 com o estabelecimento de um Conselho sobre Cartões de Passaporte. Este Conselho foi responsável por elaborar as recomendações de padronização de um passaporte ou cartão que pudesse ser lido por máquina, em função da crescente necessidade de verificação de passageiros em controles de fronteira. Em 1980, as especificações desenvolvidas pelo Conselho foram publicadas na primeira edição do Doc 9303, nomeado como Machine Readable Travel Documents.

Em 1984, a ICAO estabeleceu a hoje conhecida como Technical Advisory Group on Machine Readable Travel Documents (TAG/MRTD), composta por oficiais do governo que se especializaram na emissão e inspeção de passaportes e outros documentos de viajem, para atualizar e melhorar as especificações que foram inicialmente criadas pelo Conselho.

Em 1998, o New Technologies Working Group do TAG/MRTD começam a estabelecer um sistema de identificação biométrica mais eficiente e associado aos meios de armazenamentos utilizadas pelas aplicações nos MRTD, particularmente em relação a emissão de documentos e considerações de imigração. A maior parte do trabalho foi completada durante os eventos ocorridos em 11 de Setembro de 2001 nos Estados Unidos, resultando em uma maior importância na segurança dos documentos de viajem e na identificação do titular. Os resultados foram publicados, em Setembro de 2006, no volume 2 da sexta edição do Doc 9303.

1.1 Custo benefício

Conforme o aumento do fluxo de passageiros aumentar e mais países focarem na otimização da autenticação com sistemas automatizados, o MRTD irá desempenhar um papel central nestes sistemas mais modernos. Equipamentos para leitura e acesso aos bancos de dados serão um alto investimento inicial, mas o retorno pode ser esperado pelos aumentos em segurança, automatização, velocidade e confiabilidade nas verificações que o novo sistema irá proporcionar. O uso de MRTD também pode tornar possível os países eliminarem a necessidade da inspeção de documentos em papel e os custos administrativos envolvidos com o processo manual.

1.2 Passos da autenticação

Na prática, o portador do MRTD simplesmente apresenta seu ePassport para leitura e rapidamente o sistema confirmaria a identidade do portador, comparando os dados contidos no chip, impressos no documento e os dados de biometria do titular.

Internamente, o MRTD possui os seguintes passos:

- Conexão do cartão com a base leitora (com ou sem contato)
- Leitura do código MRZ ou inserção manual dos dados impressos
- Autenticação do código e permissão de acesso à leitura
- Leitura dos grupos de dados permitidos
- Verificação da autenticidade do conteúdo
- Confirmação e impressão dos dados

O apêndice A - Passos na autenticação MRtd, possui o diagrama completo das atividades de cada passo. Já o detalhamento técnico é abordado na seção 6.1, onde também são feitos comentários complementares sobre as vulnerabilidades relacionadas (figura 4.3).

1.3 Características

1.3.1 Machine Readable Zone (MRZ)

Os MRTDs produzidos de acordo com o Doc 9303 incorporam um código MRZ para facilitar a inspeção de documentos e reduzir o tempo necessário nos processos administrativos. O MRZ é destinado a dados para uso internacional em conformidade com os padrões estabelecidos para MRTDs.

Em consideração com as leis nacionais de privacidade, os dados contidos no MRZ devem ser visíveis tanto a olho nu como por máquinas. O dados devem obedecer um padrão em comum de forma que todos os leitores em conformidade com o Doc 9303 possam reconhecer os caracteres e se comunicar em um protocolo padrão (ex.: ASCII) que será compatível com a infraestrutura necessária definida por cada País.

1.3.2 Estrutura Lógica de Dados

Esta seção define uma Estrutura Lógica de Dados (Logical Data Structure, LDS) necessária para a operação global dos MRTDs.

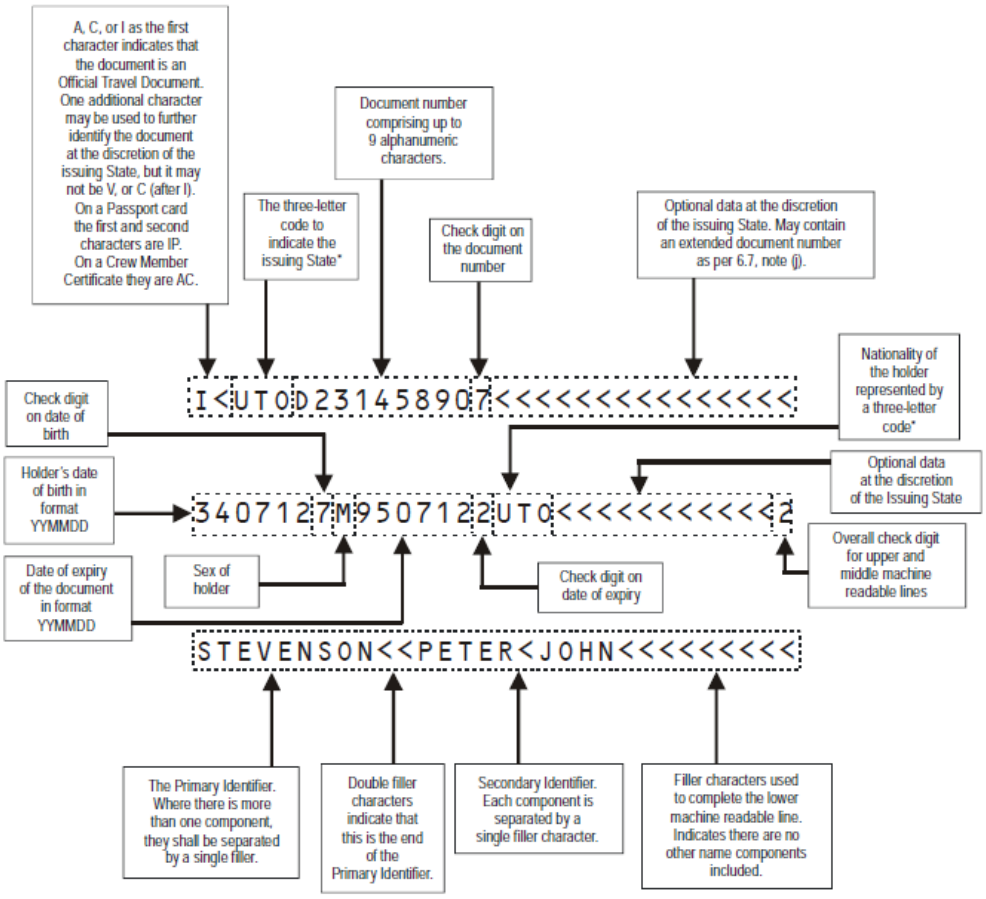

Figura 1 – Estrutura de informações do MRZ

1.3.2.1 Dados opcionais e obrigatórios

Uma série de Dados opcionais e obrigatórios foram definidos para que a LDS cumpra os requisitos globais de processamento por MRTDs, exibidos na figura abaixo.

1.3.2.2 Ordenação e Agrupamento dos Dados

Os dados são agrupados por relação e obrigatoriedade. Estes grupos são então ordenados logicamente como vistos na figura anterior. Esta ordem lógica foi padronizada para atender os requisitos globais para facilitar e aumentar a segurança durante o processamento de MRtds.

Quatro tipo de dados são obrigatórios e devem sempre estar inclusos se um LDS for criado com a tecnologia de circuito integrado sem fio:

- os que definem o conteúdo da machine readable zone (MRZ) do MRtd (DG1)
- imagem facial do titular do MRtd
- o arquivo EF.COM, que contém a versão e lista de tags

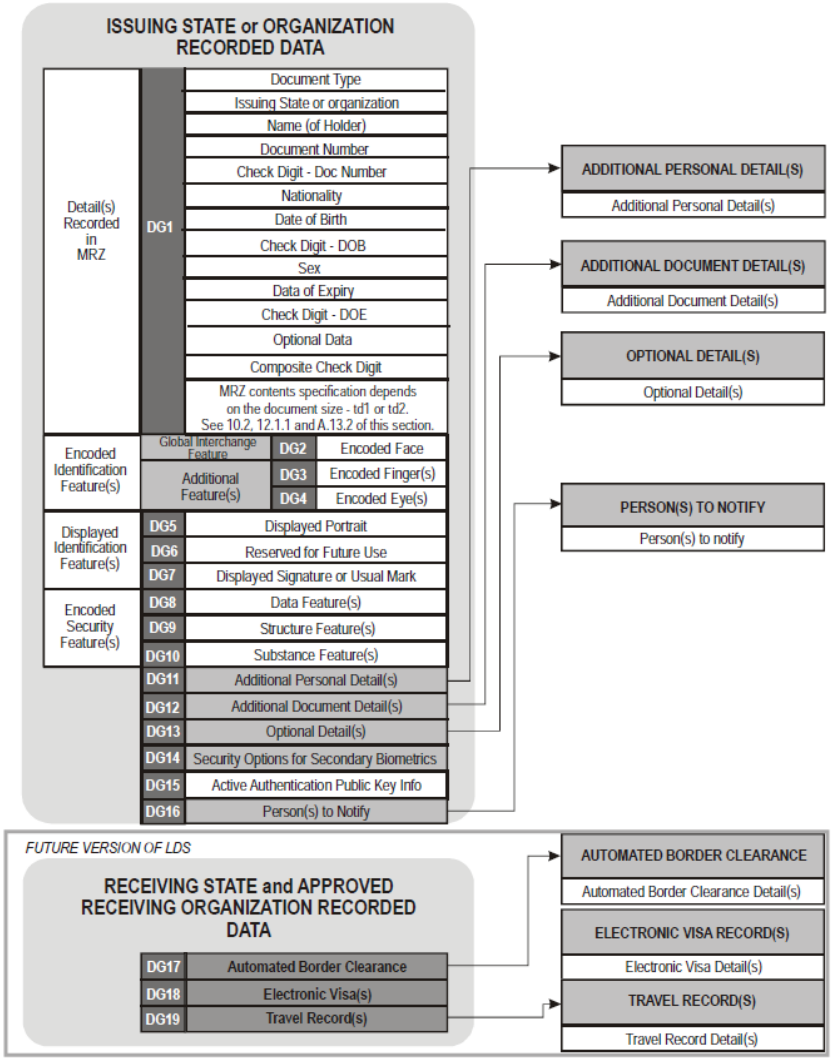

Figura 2 – Grupos de Dados do LDS

- o arquivo EF.SOD, que garante a integridade e autenticidade dos dados

Todos os outros dados definidos por um País ou organização são opcionais. O agrupamento destes Dados adicionais podem ou não estarem presentes em uma LDS. Ele é considerado uma entidade única e coesiva que contém o número total de grupo de Dados gravados e é usado durante a leitura do MRtd. Dentro do LDS, grupos lógicos de dados foram estabelecidos. Estes grupos lógicos serão referenciados como Grupo de Dados.

Cada Grupo de Dados é associado a um número de referência. A figura 2 exibe os números de referências de cada Grupo de Dados, por exemplo, "DG2"identifica o Grupo de Dado 2, que possui as informações biométricas faciais do titular.

A ICAO reconhece que os Países membros podem optar pela verificação adicional através de tecnologias de biometria através de reconhecimento de digitais e/ou íris, estes dados devem ser protegidos (criptografados) nos Grupo de Dados 3 (DG3) e Grupo de Dados 4 (DG4) respectivamente.

1.4 Autenticação dos Dados

Os mecanismos de autenticação são executados na seguinte ordem:

1.4.1 Controle de Acesso Básico (BAC)

Opcional.

Ao comparar um MRtd equipado com tecnologia sem contato e um MRtd tradicional, notam-se duas diferenças:

- Os dados armazenados no chip sem contato podem ser lidos eletronicamente sem autorização de leitura do documento (roubo de dados).
- Um canal de comunicação não criptografado entre o chip sem contato e o leitor pode ser interceptado a alguns metros de distância.

Embora existam várias medidas físicas possíveis contra o roubo de dados, nenhuma destas se aplicam a interceptação à distância. Portanto, é altamente recomendado que os Países implementem o mecanismo de Controle Básico de Acesso (Basic Access Control, BAC) para por exemplo, que o dono do MRtd tenha ciência de que os dados em seu documento estejam sendo lidos de forma segura. O mecanismo BAC deve prevenir tanto o roubo como a interceptação dos dados.

Esta prática é recomendada com a intenção de proteger a privacidade e atender aos direitos dos portadores. Este mecanismo de controle de acesso porém, é opcional. Se implementado, ele deve assegurar que o conteúdo somente seja lido depois do consentimento do titular do MRtd.

Um chip que é protegido pelo mecanismo BAC nega acesso ao seu conteúdo a não ser que o sistema de verificação prove que está autorizado ao acesso. Esta prova é dada através de um protocolo de desafio-resposta, onde o sistema demonstra conhecimento das chaves de acesso particulares do documento BAC, que são derivações da informação do campo MRZ.

O sistema de inspeção deve fornecer esta informação antes de ser habilitado a ler os dados do MRtd. A informação deve ser recuperada da leitura ótica/visual do MRtd (no campo MRZ). Também deve ser possível ao sistema inserir manualmente estes dados caso o leitor ótico para o MRZ não esteja disponível.

Adicionalmente, após a autenticação bem sucedida do sistema, é necessário que o MRtd criptografe o canal de comunicação entre o leitor e o MRtd através de técnicas de troca segura de mensagens. Com a criptografia do canal de comunicação, uma interceptação e decodificação dos dados requerem um esforço adicional considerável e inviável.

1.4.2 Controle de Acesso Estendido (EAC)

Opcional.

O mecanismo opcional de controle de acesso estendido (Extended Access Control, EAC) é similar ao BAC descrito anteriormente. Portanto, outra chave de acesso ao documento estendida é necessária (Document Extended Access Key), em complemento as chaves de acesso ao documento BAC.

1.4.3 Autenticação Passiva (PA)

Obrigatória.

Em adição aos Grupo de Dados LDS, o MRtd também contém um Document Security Object (SOD). Este objeto é assinado digitalmente pelo País emissor e contém representações em hash dos conteúdos LDS.

A autenticação passiva prova que o conteúdo do SOD e LDS são autênticos e inalterados, mas ele não previne uma cópia exata do conteúdo do MRtd ou substituição do chip. Portanto, o sistema de autenticação passiva deve ser complementada por uma inspeção física do MRtd.

1.4.4 Autenticação Ativa (AA)

Opcional.

Um País emissor deve proteger seus MRtds contra clonagem. Isto pode ser feito implementando o mecanismo de Autenticação Ativa. Se suportado, a Autenticação Ativa deve assegurar que o chip não foi substituído, através de procolos desafio-resposta entre o sistema verificador e o chip do MRtd.

1.4.5 Segurança biométrica adicional

Os dados pessoais armazenados no chip do MRtd definidos como obrigatórios como o mínimo necessário para a interoperabilidade global são o MRZ e a imagem digitalizada do rosto do titular. Ambos devem podem ser vistos (lidos) após o MRtd ser aberto e oferecido a uma inspeção. Apesar da imagem facial ter sido definida como a biometria primária, a ICAO endossou também o uso de imagens das impressões digitais e/ou íris. Cada País pode escolher armazenar e/ou definir acesso limitado ou criptografado a estes dados, definições estas a serem decididas por cada país.

O acesso aos dados pessoais mais particulares deve ser mais restrito. Isto pode ser realizado de duas formas: controle de acesso estendido ou criptografia dos dados. Apesar desta opções citadas, a ICAO não propõe ou especifica qualquer padrão ou práticas de

desenvolvimento nestas áreas. A definição do algoritmo de codificação/decodificação e as chaves a serem utilizadas cabe a cada país definir e implementar.

1.5 PKI para MRTDs

Esta seção fornece especificações utilizadas pelos Países e fabricantes para um sistema de autenticação envolvendo estruturas de chaves públicas (PKI) para uso de assinaturas digitais para MRTds. As especificações porém, não definem uma implementação completa da estrutura de uso de PKIs para cada país. Funcionalidades adicionais podem ser inseridas de acordo com a intenção de cada órgão emissor.

Apesar do uso das técnicas de criptografia de chave pública adicionar complicações na implementação de MRTds, estas técnicas incrementam a segurança nos pontos de controle com medidas adicionais de verificação de autenticidade. Mas é importante ressaltar que somente o uso de tais técnicas para determinar a autenticidade de um MRtd não devem ser o único fator conclusivo.

No caso em que os dados de um documento não possam ser utilizados, seja como resultado de um certificado revogado, verificação de assinatura inválida, ou o chip esteja vazio, o MRtd não é necessariamente invalidado. Neste caso, o País deve verificar os outros recursos de segurança do documento para prosseguir a validação.

1.5.1 Conceitos Gerais

Os princípios dos sistemas de PKI evoluíram seu uso até se tornarem altamente complexas em suas aplicações. Seu uso primário foi em transações eletrônicas, onde as chaves eram confiadas a um grande número de usuários e agências, o que resultou em um sistema elaborado de chaves, onde as chaves públicas são emitidas em "certificados"que são assinados digitalmente por organizações confiáveis chamadas Autoridades Certificadoras (certificate authorities, CAs).

A confiança nestas organizações é ainda verificada por CAs em um nível hierárquico mais alto, cada uma emitindo a chave e assinando o certificado para o órgão abaixo na hierarquia. O nível mais elevado é chamada "Root CA". Hierarquias diferentes podem cruzar certificações para estabelecer uma maior confiança pelas chaves emitidas entre elas.

Um fator complicante é a necessidade da Lista de Certificados Revogados (CRLs), indicando quando uma chave (certificado) perdeu sua validade independente do motivo. Na verdade, ao revogar um certificado e publicá-lo nesta CRL, o emissor do certificado informa aos destinatários que o conteúdo não pode mais ser confiado. A necessidade de verificar certificados para cada e toda transação implica em acessos múltiplos aos registros de CA e CRL em base de dados diferentes.

Em caso de uma chave ser comprometida, um mecanismo de prevenção deve ser utilizado para avisar os outros Países para verificaram estes documentos (ainda válidos por um período) com maior cautela.

1.5.2 País Emissor (Country Signing CA)

A hierarquia CA, na qual a geração de chaves será incorporada, é relevante somente nesta seção por envolver os certificados que são distribuídos entre os Países destinatários. O maior nível de certificado que for distribuído deve agir como um meio de confiança para o País receptor. Neste trabalho, este certificado é referenciado como Country Signing CA Certificate (CCSCA). O CCSCA deve ser auto-assinado e emitido pelo Country Signing CA (CSCA). É recomendado que os pares de chave CSCA (KPrCSCA) sejam gerados e armazenados em uma estrutura altamente protegida e off-line pelo País emissor. Os certificados dos Países CCSCA devem ser distribuídos por meios estritamente diplomáticos. Cada CCSCA gerado por cada País deve ser encaminhado para a ICAO, para validação do assinante de certificado (Document Signer Certificates, CDS). O par de chaves CSCA é utilizado para assinar os CDS.

1.5.3 Assinante de Documento

É recomendado que os pares de chave DS (Document Signer Key Pairs, KPrDS) sejam gerados e armazenados e protegidos pelo próprio País emissor. Cada certificado de assinatura de documento (Document Signer Certificate, CDS) gerado por cada País deve ser encaminhado para a ICAO e pode ser armazenado em um MRtd. A chave privada DS (Document Signer Private Key, KPrDS) é usada para assinar objetos segurança do documento (Document Security Objects, SOD). Cada SOD gerado por cada País deve ser armazenado em seus respectivos MRTds.

1.5.4 Revogação de Certificado

Os Países emissores podem revogar certificados no caso de um incidente (como o comprometimento de chaves). Esta revogação deve ser comunicada biliteralmente a todos os outros Países participantes e ao Diretório ICAO de Chaves Públicas dentro de 48 horas. Em caso de ausência de incidentes os Países emissores devem distribuir relatórios dos CRLs no máximo a cada 90 dias.

1.5.5 Diretório ICAO de Chaves Públicas

A fim de compartilhar eficientemente os Certificados de Assinatura de Documento (Document Signer Certificates, CDS) entre todos os Países, a ICAO irá fornecer um serviço de Diretório de Chaves Públicas (PKD). Este serviço deve aceitar informações de chaves

públicas dos Países, armazená-las em um diretório e torná-las acessíveis a todos os outros Países. O acesso para atualizar as listas de certificados armazenados deve ser restrita apenas aos Países participantes. Não há controle de acesso para leitura do PKD.

1.5.6 Country Signing CA Certificate

Os Country Signing CA Certificates (CCSCA) não fazem parte do serviço ICAO PKD. O PKD entretanto, podem utilizar o CCSCA para verificar a autenticidade e integridade dos Certificados de Assinatura de Documento recebidos por outros Países, antes de publicá-los. A ICAO não permite acesso aberto ao CCSCA.

1.5.7 Certificados de Assinatura de Documento

A ICAO PKD pretende ser o repositório de todos os Certificados de Assinatura de Documento (Document Signer Certificates, CDS) utilizados por todos os Países integrantes. Isto inclui certificados ativamente sendo utilizados a qualquer momento para assinaturas assim como para aqueles que não estejam mais sendo utilizados mas ainda ativos para os MRtds emitidos. A ICAO PKD irá ser o mecanismo primário de distribuição destes CDSs e portanto deve ser populado e mantido atualizado por todos os Países integrantes. A informação de Chave Pública de determinado País armazenado no PKD deve estar disponível a todos os outros órgãos (mesmo que não sejam Países integrantes) que precisem desta informação para validação da autenticidade dos dados do MRtd.

1.5.8 Lista de Certificados Revogados

O PKD também deverá ser um repositório para todas as Listas de Certificados Revogados (Certificate Revocation Lists, CRLs) emitidos por cada País integrante. Apesar dos Países deverem inicialmente distribuir as CRLs biliteralmente, eles devem também comunicá-las ao PKD. Assim, o ICAO PKD será um meio secundário de distribuição de CRLs.

1.6 Segurança dos dados em MRtds (sumário)

Apesar da Autenticação Passiva através de assinaturas digitais, os Países podem escolher mecanismos adicionais de segurança, utilizando outros meios mais complexos para proteger o chip MRtd e seus dados. As opções listadas na tabela a seguir podem ser combinadas para atingir níveis de segurança adicionais seguindo os padrões ISO/IEC.

Method	Issuer	Insp. System	Benefits	Deficiencies
BASELINE SECURITY METHOD				
Passive Authentication (5.6.1)	**M**	**M**	Proves that the contents of the SO_D and the LDS are authentic and not changed.	Does not prevent an exact copy or IC substitution. Does not prevent unauthorized access. Does not prevent skimming.
ADVANCED SECURITY METHODS				
Comparison of conventional MRZ(OCR-B) and IC-based MRZ(LDS)	N/A	O	Proves that contactless IC's content and physical MRtd belong together.	Adds (minor) complexity. Does not prevent an exact copy of contactless IC and conventional document.
Active Authentication (5.6.2)	O	O	Prevents copying the SO_D and proves that it has been read from the authentic contactless IC. Proves that the contactless IC has not been substituted.	Adds complexity. Requires processor-ICs.
Basic Access Control (5.7)	O	O	Prevents skimming and misuse. Prevents eavesdropping on the communications between MRtd and inspection system (when used to set up encrypted session channel).	Does not prevent an exact copy or IC substitution (requires also copying of the conventional document). Adds complexity. Requires processor-ICs.
Extended Access Control (5.8.1)	O	O	Prevents unauthorized access to additional biometrics. Prevents skimming of additional biometrics.	Requires additional key management. Does not prevent an exact copy or IC substitution (requires also copying of the conventional document). Adds complexity. Requires processor-ICs.
Data Encryption (5.8.2)	O	O	Secures additional biometrics. Does not require processor-ICs.	Requires complex decryption key management. Does not prevent an exact copy or IC substitution. Adds complexity.

Figura 3 – Métodos de Segurança do MRTD

2 Registro de Identidade Civil

O Registro de Identidade Civil é o foco deste estudo. Primeiro por ainda estar em fase de implantação (estando aberto a novas funcionalidades), e segundo por seguir os mesmos padrões determinados para o MRTD pela ICAO, diferindo apenas na possibilidade de verificação por meio de um segundo chip com contato.

Com a implantação do RIC, os cidadãos não irão mais precisar registrar legalmente as suas assinaturas em um cartório local, onde os documentos frequentemente têm que ser oficialmente reconhecidos e carimbados para se tornarem válidos. O RIC irá centralizar o registro de todos os brasileiros no governo federal central, em vez de serem registrados por cada estado.

Além de reduzir e automatizar a burocracia do Brasil, o RIC também será importante para a aplicação da lei e proteção contra fraudes. Atualmente, um criminoso no Estado do Ceará poderia ir para o Rio Grande do Sul e solicitar uma nova identidade, simplesmente pela falta ou inexistência de comunicação entre os estados.

2.1 Especificações

Segundo a resolução SE N2, de 22 de Novembro de 2011 que dispõe sobre as especificações técnicas básicas do documento de Registro de Identidade Civil, as principais especificações técnicas de interesse específico neste trabalho foram:

2.1.1 Características Físicas

- O cartão utilizado como suporte documental para o novo documento de identificação brasileiro, e que trará o número RIC (Registro de Identidade Civil), deverá atender às normas internacionais para documentos similares, em especial às normas ISO 10732 e 1831 (reconhecimento óptico de caracteres), ISO 7810 (características físicas do cartão), e Documento 9303 da ICAO (documentos de viagem de leitura mecânica)

- Dizeres indicativos dos campos dos dados variáveis (nome, sexo, nacionalidade, data de nascimento, data de validade, número RIC, documento de origem, RG/UF, CPF, NIS, título de eleitor, filiação, naturalidade, órgão emissor, local de emissão, data de emissão, observações)

- Fotografia do titular;

- Impressão datiloscópica do anular direito do titular;

- Assinatura digitalizada do titular;
- Código OCR - B na Zona de Leitura Mecânica (MRZ);
- Número de série do cartão.

2.1.2 Especificações Técnicas

- Serão embarcados dois chips no documento de Registro de Identidade Civil, um sem contato (para o RIC funcionar como um documento de viagem, padrão ICAO) e outro com contato (para questões de autenticação e suporte a multi-aplicação).
- Chip sem contato:
  - As especificações/arquiteturas do chip sem contato devem possuir características de acordo com as recomendações ICAO DOC 9303 Machine Readable Travel Documents; ISO/IEC 14443 - Contactless Integrated Circuits Cards - Proximity Cards; ICAO NTWG, Use of Contactless Integrated Circuits In Machine Readable Travel Documents Technical Report;
  - Suporte a aplicação BAC/AA e EAC.
  - Hardware com suporte a infra estrutura de chave pública/privada permitindo os algoritmos indicados no documento ICAO 9303, assim como no mínimo todas as recomendações para tamanhos míninos da chave em relação a Country Signing CA Keys, Document Signer Keys e Active authentication Keys:
  - Suporte a 3DES e AES;
  - Os algoritmos de hash devem ser os especificados no documento ICAO 9303;
  - A estrutura dos Data Groups no LDS deve seguir a padronizada no documento da ICAO 9303, contendo no mínimo os seguintes Data Groups:
  - EF.COM;
  - DG1 - MRZ (com todos os elementos de dados) - BAC/AA;
  - DG2 - Face (JPEG 2000) - BAC/AA;
  - DG3 - FingerPrint (WSQ) - EAC/AA;
  - DG14 - EAC;
  - DG15 - AA;
  - EF.SOD - Hash e Assinatura Digital.
  - A utilização de novos dados deverá seguir a especificação do documento ICAO 9303.

- O fabricante deve fornecer o teste de interoperabilidade leitor/chip da ICAO, além de comprovar sua participação no teste;

- A integridade, a autenticidade e a confidencialidade dos dados digitalmente armazenados devem estar de acordo com ICAO NTWG, PKI for Machine Readable Travel Documents Offering ICC Read - Only Acess, Technical Report;

- Chip com contato:

  – Todas as especificações/arquiteturas do chip com contato devem possuir características de ordenamento lógico de acordo com as recomendações ISO/IEC 7816 - Identification Cards, Integrated Circuit Cards; ISO/IEC 19784 - Information Technology - Biometric Application Programming Interface; ISO/IEC 19794 - Biometric Data Interchange Formats.

  – Suporte a 3DES e AES;

  – Suporte a multi-aplicação;

  – Suporte a MOC (match-on-card).

2.2 Campos de Testes

De acordo com o Ministério da Justiça, a previsão inicial é emitir dois milhões de cartões a partir de 2011 – sendo os 100 mil primeiros para a Bahia, Rio de Janeiro, Distrito Federal e para as cidades Hidrolândia (GO), Nísia Floresta (RN), Rio Sono (TO) e a Ilha de Itamaracá (PE).

Cerca de 60 mil registros serão expedidos pelo DF, RJ e BA e outros 40 mil cartões RIC ficarão a cargo dos quatro municípios selecionados. A substituição do RG atual pelo novo documento será feita de forma gradual, ao longo de nove anos.

2.3 RIC e o MRTD

Apesar do RIC obedecer aos padrões do ICAO Doc9303, foram considerados duas características principais nesta especificação do RIC:

- As características físicas do documento e do chip sem contato devem obedecer aos mesmos padrões do ePassport.

- O diferencial de possuir um chip sem contato (inexistente no ePassport) possibilita a retrocompatibilidade com outras aplicações já existes (eCPF, eCNPJ) além de facilitar a adição de novos recursos.

Por ser baseado em um modelo pré-existente e adotado mundialmente, o RIC possui potencial para ser bem sucedido em sua proposta de centralização dos dados e proteção contra fraudes.

Entretanto, por seguir exatamente o mesmo padrão, também foram incorporadas as mesmas falhas de segurança já documentadas por especialistas, sendo a grande maioria de fácil correção. Algumas destas vulnerabilidades serão abordadas no final deste estudo.

3 Implementação do protótipo de extensão do MRTD

Apesar das informações contidas no MRTD serem mais que suficientes para a identificação do titular, o potencial de processamento e armazenamento de um smartcard é subutilizado. O protótipo desenvolvido tem como objetivo explorar as possibilidades de aplicações adicionais que possam ser gravadas no mesmo cartão.

Na tecnologia javacard todas as aplicações são encapsuladas, de forma que não é possível acessar dados externos à aplicação ativa. Assim, a segurança dos dados pessoais do titular sempre estaria seguras pelo padrão MRTD independente da forma como as novas aplicações sejam projetadas.

Afim de se beneficiar do processo de autenticação do usuário, a proposta do aplicativo é de extender as informações contidas no documento, com dados não relevantes para o governo ou temporários (duração menor que a validade do documento) e quaisquer outros que sejam convenientes para o usuário.

A solução apresentada foi o uso de certificados de atributos, que apesar de utilizar uma infraestrutura de chaves públicas, complementa os dados do MRTD de forma simples e confiável. Com o certificado de atributo gravado no mesmo documento que identifica o usuário, o vínculo dos novos dados é direto, facilitando a verificação.

3.1 Certificado de atributos

O certificado de atributo é uma estrutura de dados de segurança e identificação, constantes em campos de um certificado digital, ou anexadas a um outro certificado e assinados com a chave pública da autoridade que o emitiu. Esse certificado traz informações adicionais sobre seu titular, como cargo, função, profissão etc. O certificado de atributo segue o padrão X.509, adotado pela ICP-Brasil na emissão de certificados de pessoa física, jurídica e de equipamentos.

Em uma analogia, uma chave pública possui a funcionalidade de um passaporte: identifica o titular, possui um período de validade e só pode ser emitido através de agentes específicos após várias verificações. Já um certificado de atributo é como o visto do passaporte: é normalmente emitido por autoridades diferentes e possui uma validade curta. Como o pedido de um visto requer um passaporte, o processo é bem mais simples.

Informações de autorização podem ser estendidas a uma chave ou vinculadas na forma de um certificado de atributo (CA). A primeira opção não é recomendável por dois

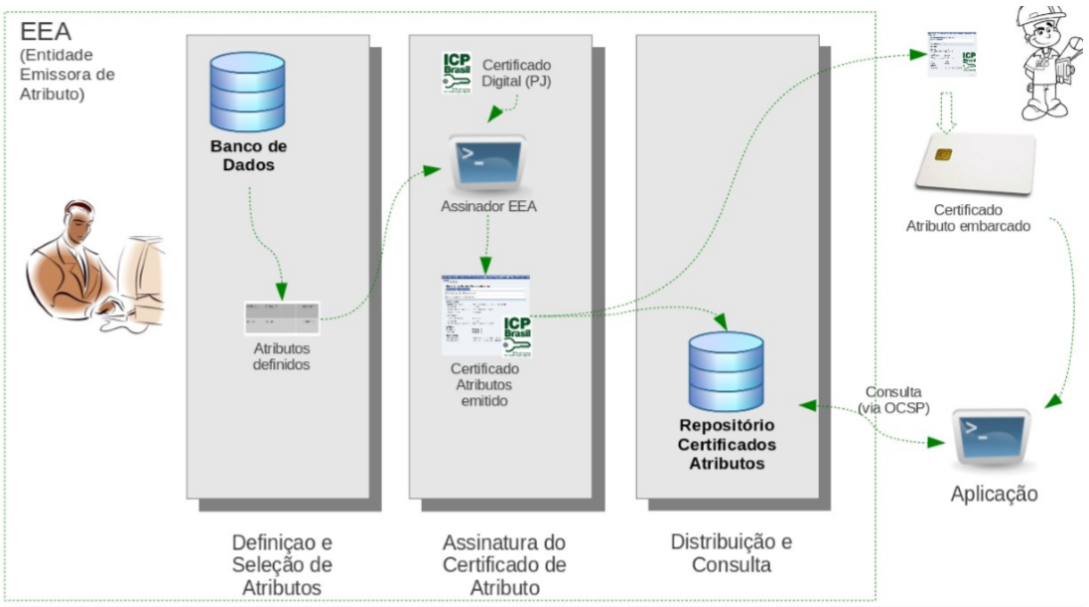

Figura 4 – Ciclo de vida de um Certificado de Atributo (ICP-Brasil, 2012)

motivos: diferentes tempos de validade e relacionamento direto entre as duas para emissão.

Caso a informação de autorização seja uma estensão de uma chave, é provável que o tempo de vida da chave seja reduzido. Além disso, durante a emissão da chave o emissor precisaria consultar os órgãos responsáveis por cada autorização, complicando ainda mais o processo de emissão da chave. Por estes motivos, geralmente é melhor separar as informações de autorização da chave pública.

Um CA pode ser utilizado em vários serviços de segurança, como controle de acesso, autenticidade de conteúdo e não-repúdio sem interferir nas chaves originais.

3.2 Sistema Gerenciador de Certificado de Atributos (SGCA

O SGCA é um aplicação (web ou standalone) a partir da qual usuários poderão efetuar o pedido para a emissão do seu certicado digital de atributo, onde um operador responsável pelo software poderá aprovar ou negar o pedido.

Os certificados de atributos aprovados, podem ser copiados pelos respectivos usuários, além de consultar a validade e verificar o status do certicado. O desenvolvimento desse software possibilitará maior intercâmbio de informacões e conhecimentos sobre o uso do certicado de atributo. Também poderá servir como fonte de autenticação de usuários, utilizando a conexão direta com um servidor LDAP (Lightweight Directory Access Protocol).

Neste estudo, foi desenvolvido um protótipo que grava um certificado de atributo validando um estudante como aluno de uma instituição de ensino. O certificado seria emitido

pela própria instituição responsável (gravado em um RIC, por exemplo) e a validação do certificado poderia ser feito por outras entidades de forma confiável e automatizada.

4 Definição do problema

Para melhor compreensão dos procedimentos de autenticação e segurança nos padrões ICAO (utilizado como referência pelo ePassport e RIC) foi utilizado como base uma implementação da JMRTD.org do MRTD, uma aplicação de código-aberto em Java seguindo toda a especificação da ICAO.

A centralização de várias informações em um único cartão e a possibilidade de validação automatizada é de evidente praticidade e conveniência. Esta grande responsabilidade em um único cartão demanda igual confiabilidade e segurança. Este estudo teve como principal objetivo responder a questões como:

- O RIC é totalmente seguro contra fraudes?
- Como é garantido o acesso autorizado à leitura do cartão?
- Em caso de perda/roubo, como os dados serão protegidos?
- Em caso de dano ou defeito no cartão, a autenticação ainda é possível?
- Qual a viabilidade técnica real?
- Quais funcionalidades adicionais podem ser incluídas?

Baseado em cada falha de segurança encontrada e outras documentadas por especialistas, as respectivas causa, impacto e soluções serão descritos na seção 6 deste documento.

4.1 JMRTD

O projeto Java Machine Readable Travel Document possui uma aplicação cliente presente no cartão ("passaport applet") e uma aplicação servidor com a API de acesso aos ePassports, que implementa um sistema de inspeção (leitura, decodificação e validação das informações do cartão) e emissão de passaportes de acordo com os padrões ICAO.

O JRMTD foi desenvolvido inicialmente para simular e validar a segurança das especificações do ePassport, implementando tudo conforme especificado pelo DOC9003 principalmente:

- Certificados PKD e CSCA
- Extended access control (EAC)

- Logical Data Structure (LDS)

4.2 Procedimentos de Autenticação

Com base no código fonte do JMRTD e especificação do Doc 9303, os seguintes passos são seguidos durante uma autenticação de ePassport:

1. Conexão do cartão com a base leitora
    - Leitura do código MRZ ou inserção manual dos dados impressos
    - Conexão com o chip com ou sem contato
    - Autenticação do código
2. Permissão de acesso à leitura (BAC e EAC)
    - Leitura dos grupos de dados permitidos
3. Verificação da autenticidade do conteúdo (PA e AA)
4. Confirmação e impressão dos dados

Alguns destes métodos de autenticação são opcionais. No caso da verificação BAC (obrigatória no JMRTD) falhar, nenhum conteúdo será exibido (acesso negado). Já se o EAC (opcional) não estiver disponível, o JMRTD não tem acesso apenas aos grupos de dados DG3 e DG4 (biometria).

Uma falha do EAC não significa que há algo errado com o cartão, indica apenas que não há permissão de acesso a estes dados mais sigilosos. O propósito do EAC é permitir apenas que sistemas com credenciais específicas (PK e certificado assinado pelo país emissor) possam ter acesso aos dados biométricos.

No passo 3, a autenticação passiva (PA) é obrigatória para validar os dados contidos no cartão. Já a autenticação ativa (AA), apesar de opcional, é altamente recomendada afim de evitar a clonagem.

4.3 Conexão com a base leitora

4.3.1 Leitura do código MRZ

A leitura ótica do código MRZ extrai os principais dados do documento:

- Número do documento = 123456789-7

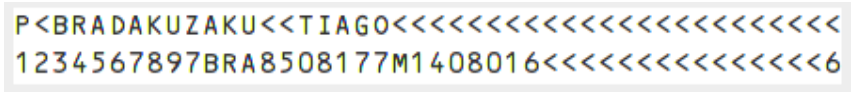

Figura 5 – Exemplo de código MRZ

- Data de nascimento = 850817-7
- Data de expiração = 140801-6
- MRZ info = 123456789785081771408016

Apenas os 16 bytes mais significantes do hash SHA-1 desta informação são utilizados para derivar ("key seed") as chaves BAC.

4.4 Controle Básico de Acesso (BAC)

Opcional, mas extremamente recomendado.

4.4.1 Autenticação do código

A computação do par de chaves 3DES derivadas de um "key seed"(K_{seed}) é utilizada tanto na criação das chaves BAC (K_{ENC} e K_{MAC}) como nas chaves de sessão para comunicação segura.

Um contador c de 32 bits é utilizado em combinação com a Kseed para derivar as chaves BAC. Dependendo de qual chave a ser gerada (encriptação ou MAC), os seguintes valores deve ser utilizados:

D = "239AB9CB282DAF66231DC5A4DF6BFBAE00000001"

- $c = 1$ (i.e. '0x 00 00 00 01') para chave de encriptação (K_{ENC}).
- $c = 2$ (i.e. '0x 00 00 00 02') para chave MAC (K_{MAC}).

Os seguintes passos são seguidos para derivar os pares de chaves 3DES:

1. Concatena-se o K_{seed} com c: $D = K_{seed} + c$.
   D = "239AB9CB282DAF66231DC5A4DF6BFBAE00000001"
2. Calcula-se o hash: $H = SHA1(D)$.
   HSHA-1(D) = "AB94FCEDF2664EDFB9B291F85D7F77F27F2F4A9D"
3. Os primeiros 8 Bytes de H formam a chave K_a e os próximos 8 Bytes a chave K_b.
   Ka = "AB94FCEDF2664EDF"
   Kb = "B9B291F85D7F77F2"

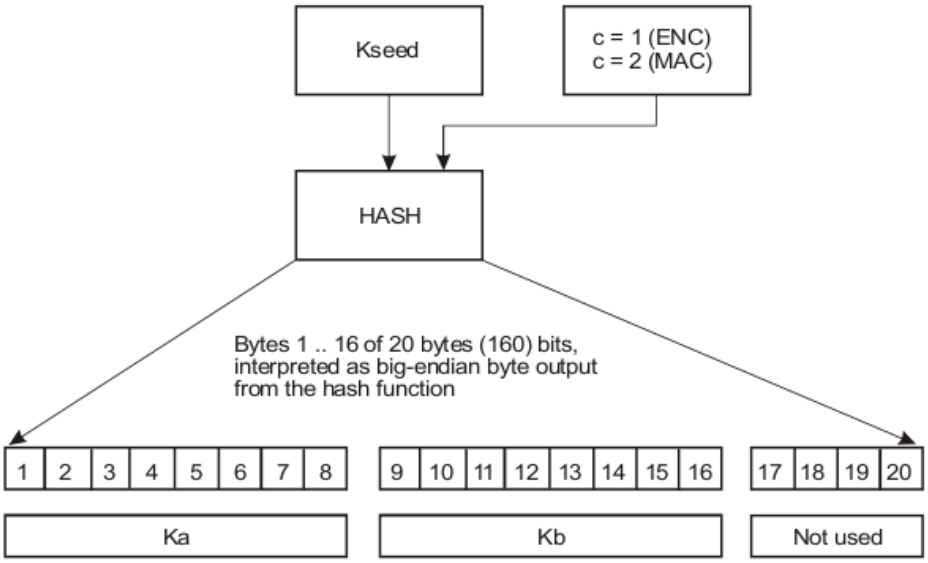

Figura 6 – Computação das chaves BAC

4. Ajuste dos bits de paridade das chaves K_a e K_b para formar chaves DES.
   Ka = "AB94FDECF2674FDF"
   Kb = "B9B391F85D7F76F2"

4.4.2 Autenticação e criação das chaves

A autenticação e criação da chave é definida por um protocolo de três passos desafio-resposta, de acordo com o a ISO/IEC 11770-2, utilizando 3DES como cifrador. Um código de verificação (checksum) é calculado de acordo com a ISO/IEC 9797-1 e concatenado ao texto cifrado.

Os seguintes passos são realizados durante a autenticação entre o IFD e o ICC:

1. O IFD solicita um desafio ao ICC através do comando GET CHALLENGE. O ICC gera uma resposta retornando seu RND_{ICC}.
2. O IFD realiza as seguintes operações:
   a) Gera seu próprio RND_{IFD} e chave K_{IFD}.
   b) Faz a concatenação: $S = RND_{IFD} + RND_{ICC} + K_{IFD}$.
   c) Encriptação com a chave K_{ENC}: $E_{IFD} = E[K_{ENC}](S)$.
   d) Cálculo do checksum com a chave K_{MAC}: $M_{IFD} = MAC[K_{MAC}](E_{IFD})$.
   e) Envio do comando MUTUAL AUTHENTICATE com os dados gerados E_{IFD} e M_{IFD}
3. O ICC realiza as seguintes operações:

a) Verifica o checksum M_{IFD} sobre o E_{IFD}.

b) Decriptação de E_{IFD} .

c) Extrai o RND_{ICC} de S e o compara com o retornado pelo IFD.

d) Gera a chave K_{ICC}.

e) Faz a concatenação: $R = RND_{ICC} + RND_{IFD} + K_{ICC}$.

f) Encriptação com a chave K_{ENC}: $E_ICC = E[K_{ENC}](R)$.

g) Cálculo do checksum com a chave K_{MAC}: $M_{ICC} = MAC[K_{MAC}](E_{ICC})$.

h) Envio da resposta com os dados gerados E_{ICC} e M_{ICC}.

4. O IFD realiza as seguintes operações:

a) Verifica o checksum M_{ICC} sobre o E_{ICC}.

b) Decriptação de E_{ICC}.

c) Extração do RND_{IFD} de R e verifica se o ICC retornou o valor correto.

Um exemplo prático destes passos é apresentado no Apêndice B - Autenticação BAC no JMRtd versão 0.4.9.

4.4.3 Comunicação segura

Após uma execução bem sucedida da autenticação entre o IFD e ICC, as chaves de sessão K_a (encriptação) e Kb_b (descriptação) podem ser utilizadas. Toda comunicação subsequente deve ser protegida por comunicação segura utilizando estas chaves BAC.

4.5 Controle Estendido de Acesso (EAC)

Opcional.

A implementação deste método de criptografia cabe a cada país definir. Esta chave deve consistir tanto de chaves simétricas (sejam derivadas do MRZ ou um par de chaves assimétricas correspondentes com algum certificado verificável). O EAC requer capacidade de processamento do chip MRtd.

4.6 Verificação da autenticidade do conteúdo

4.6.1 Autenticação Passiva (PA)

Obrigatório.

Para garantir a integridades dos dados, o MRtd contém um Document Security Object (SOD) contendo as representações hash dos conteúdos LDS. Este objeto é assinado digitalmente pelo país emissor.

Assim, um sistema de verificação contendo o Document Signer Public Key (KPuDS) de cada País, poderá verificar o SOD. Autenticando as informações do SOD, todo o conteúdo do LDS também é autenticado.

O sistema realiza os seguintes passos:

1. O Document Security Object (SOD) é lido do IC.
2. O Document Signer (DS) extraído do SOD.
3. A assinatura digital do SOD é verificada pelo sistema, utilizando a Document Signer Public Key (KPuDS). O Document Signer Certificate (CDS) correspondente para esta chave foi transferido da ICAO PKD.
4. Antes de utilizar o CDS para a verificação, o sistema também deve checar a assinatura digital utilizando a Country Signing CA Public Key (KPuCSCA).
5. O sistema lê os Grupos de Dados relevantes do LDS.
6. Efetuando o hash dos conteúdos e comparando com o resultado de hash encontrado no SOD, é assegurado que o conteúdo de cada Grupo de Dado é autêntico e inalterado.

Este mecanismo de verificação não requer capacidade de processamento do chip no MRtd. Portanto é chamada de "autenticação passiva"do conteúdo do cartão.

4.6.2 Autenticação Ativa (AA)

Opcional, mas altamente recomendado.

O mecanismo de Autenticação Ativa assegura que as informações lidas são de um cartão genuíno e que os dados correspondem ao impresso no documento.

O código MRZ é lido (caso não tenha sido pelo procedimento BAC) e é comparado ao valor lido do Grupo de Dados 1 (DG1). Como a autenticidade e integridade do DG1 já foi verificada pela Autenticação Passiva, é assegurado que o código MRZ lido corresponde ao documento.

Para garantir que o Document Security Object (SOD) não é uma cópia, o chip contém seu próprio par de chaves de Autenticação Ativa (Active Authentication Key pair, KPrAA e KPuAA).

1. Uma representação hash do Data Group 15 (KPuAA) é armazenada no SOD.

2. Este hash é autenticado pela assinatura digital do emissor do documento.

3. A chave privada correspondente (KPrAA) é armazenada na memória do chip do MRtd.

Ao autenticar visualmente o MRZ (através do hash MRZ no SOD), em combinação com a resposta do desafio e finamente utilizando os pares de chaves de Autenticação Ativa, o sistema de inspeção verifica que o SOD foi lido e armazenado em um MRtd genuíno.

Um exemplo do uso prático do JMRTD é apresentado no apêndice B. Nesta simulação, todos os passos da autenticação BAC são apresentados.

4.7 Protótipo Applet SGCA

Uma nova funcionalidade proposta neste estudo é a extensão das informações contidas no RIC através de certificados de atributos. A escolha se deve a facilidade de implementação de um protótipo, já que o RIC possui um chip com contato (hardware acessível) e sua principal característica de não interferir no conteúdo original do documento.

Dados adicionais poderiam ser inclusos no RIC através dos grupos de dados não utilizados - o DG11 foi criado com este propósito (seção A.111.6 do DOC9303 pt.3 vol.2), mas pelos motivos explicados na seção 5.1, qualquer atualização nestes dados iria requerer uma requisição a um órgão responsável do governo, que pudesse validar a alteração no grupo de dado.

O certificado de atributo adiciona informações pertinentes ao titular (profissão, escolaridade, doenças, etc) de forma mais flexível. Cada instituição assume a responsabilidade pela emissão do certificado, que pode ser gravado no RIC já no próprio local e de forma segura.

Na prática, um estudante ao apresentar seu RIC comprovaria sua identidade e, com o certificado de atributo da instituição de ensino, garantiria seu vínculo estudantil. Importante ressaltar que o estabelecimento precisa de uma conexão com o SGCA para validar o certificado contido no cartão.

O apêndice C apresenta o código do applet a ser instalado no RIC, com os métodos de acesso e atualização do certificado de atributo.

O apêndice D é um exemplo simples de acesso ao java card com o applet instalado, testando o acesso ao certificado atual, gravação de um novo certificado e a confirmação da atualização.

5 Vulnerabilidades

Todas as vulnerabilidades listadas a seguir foram baseadas na especificação da ICAO Doc 9303. Algumas delas, baseadas em implementações opcionais ou em leis específicas de alguns países, podem não se aplicar ao RIC. Nestes casos, comentários sobre o RIC foram adicionados ao final de cada seção.

5.1 Controle de Acesso Básico

Para prevenir uma conexão não autorizada, ela deve ser criptografada. A chave utilizada para iniciar esta conexão é gerada a partir da leitura do código MRZ (seção 4.4.1). Por ser opcional, caso um cartão não implementar a criptografia, os dados contidos em DG1 e DG2 serão armazenados na forma de texto simples.

Para se ter acesso aos dados internos do chip do cartão, o nome do titular, data de nascimento, data de expiração e número do documento são necessários (seção 4.3). Estes dados, impressos explícitamente na segunda linha do código MRZ, não são de forma alguma secretos.

A proteção do BAC contra acesso não autorizado (através da conexão sem contato) se limita apenas ao caso do atacante não ter contato visual com o documento.

Portanto, é possível que pessoas não autorizadas também possam gerar as chaves de acesso aos dados do chip, desde que tenham estes dados básicos. Todos os hotéis, lojas ou bancos que eventualmente armazenem fotocópias do documento por segurança, possuem estas informações.

5.2 Controle de Acesso Estendido

O EAC foi criado para restringir o acesso aos grupos de dados DG3 e DG4 (biometria). Esta proteção é feita criptografando o SOD. Para decriptografar, uma infraestrutura de chave pública (PKI) precisa ser estabelecida para análise global, por todos os países associados para garantir interoperabilidade.

RIC: com a centralização pretendida pelo governo federal, a criação e manutenção de uma CRL brasileira é muito mais simples. Mas por ser um requisito opcional (seção 1.5), não há garantias de sua implementação.

5.3 Autenticação Passiva

A fim de evitar a manipulação dos grupos de dados, o sistema inspetor precisa verificar a assinatura da chave pública do documento (contida no arquivo EF.SOD) através de uma lista de certificados do país de origem (CSCA). Esta lista de certificados está longe de estar completa e validada.

O Doc 9303 define que a ICAO PKD não irá fornecer o serviço de repositório de CCSCAs (seção 1.5.5). Enquanto uma fonte confiável de certificados de assinatura dos países não existir, criminosos poderão forjar ou adulterar estes grupos de dados sem serem detectados.

Além disso, apenas com certificados válidos é possível ler os dados de DG3 e DG4. Não está claro como a Lista de Certificados Revogados (CRL) serão processados e onde devem ser armazenados. A ICAO também não se responsabilizou por este repositório (seção 1.5.8).

Um ataque pode simular seu próprio país (chaves auto-assinadas) e autenticar qualquer chip de ePassport. Na prática, os sistemas de inspeção não possuem meios de validar a autenticidade do certificado de um país desconhecido.

RIC: como existe somente um país para assinar os arquivos EF.SOD, esta vulnerabilidade não se aplica.

5.4 Autenticação Ativa

O arquivo "EF.COM"que possui o índice dos arquivos disponíveis não é protegido - priorizando desempenho no acesso aos dados. Uma cópia de todos os arquivos poderia ser feita, seguido de uma alteração neste arquivo de índice, desabilitando o próprio mecanismo de autenticação (que é opcional). Esta falha é descrita pela própria ICAO no suplemento 7 do Doc 9303 (R1-p1_v2_sIV_0006).

A falha foi "rejeitada"apesar de uma solução ter sido apresentada: utilizar o arquivo EF.SOD que é assinado, ao invés do EF.COM. Se o mecanismo AA for desabilitado, uma clonagem de chips não pode ser detectada. Além disso, esta forma de ataque pode desabilitar outros recursos de segurança opcionais, como o EAC.

A ICAO recomenda que o arquivo EF.COM não seja a única forma de validação e que o EF.SOD deve ser usado como complemento. Por ser apenas uma "recomendação", cada país tem a responsabilidade de ficar atento a esta falha grave e implementar a validação adicional.

5.5 Sistema de Inspeção

As características dos documentos são claramente especificadas pela ICAO. Já os sistemas de inspeção são responsabilidade de cada país, de forma que não há um padrão a ser seguido. Não foi criado um órgão independente que defina, documente e inspecione as normas de segurança. Sem um padrão, não há realmente como garantir a qualidade dos softwares implementados.

6 Análise

O projeto MRTD foi baseado em uma necessidade real da centralização dos certificados e automação dos processos de validação. A ICAO faz parte da ONU e é responsável pelo desenvolvimento dos padrões internacionais dos passaportes (Convenção de Chicago de 1944).

O objetivo principal é o aumento na segurança dos documentos, mas devido a complexidade dos mecanismos (BAC, EAC, PA, AA), várias vulnerabilidades também foram criadas. Como cada procedimento complementa o outro, a não implementação de um procedimento opcional pode comprometer a segurança de todo o resto.

Com base nos resultados encontrados no estudo prático do MRTD, as respostas das principais questões sobre o RIC foram:

6.1 Segurança contra fraudes

Uma cópia digital é sempre idêntica a original (ao contrário da analógica) e portanto não pode ser distinguida da original. Já uma fotocópia pode ser facilmente detectada por um inspetor. Mas é possível assumir que a longo prazo, com o crescente aumento na necessidade de automatizar os controles de acesso, que a inspeção humana seja eliminada para a maioria dos casos.

Exemplo: um passageiro com um chip clonado (válido) se aproxima do sistema automatizado de inspeção. Antes de sua viajem, ele destruiu o chip RFID original no forno microondas (inutilizando-o sem evidências), e inseriu um novo chip falso no passaporte. O chip clonado é uma cópia do chip original do passaporte mas com adulterações simples (ex.:foto do titular).

Nenhum humano poderia ver nada suspeito, e o chip adulterado permitiu a entrada do passageiro através do controle de fronteira. Esta falha já foi demonstrada por Jeroen van Beek, na conferência BlackHat USA em 2008.

Esta fraude se baseia na falta dos mecanismos opcionais. Portanto, é de extrema importância o RIC implementar todas as definições afim de evitar fraudes básicas como esta.

6.2 Acesso autorizado à leitura do cartão

A vulnerabilidade dos chips RFID é amplamente discutida em todas as áreas. O acesso não autorizado envolveria um leitor de cartão oculto, que pode capturar dados até

10cm - e esta distância pode ser aumentada até 75cm com energia e antenas adequados.

Não existe uma solução sugerida pela ICAO. A alternativa inicial de se utilizar um código de acesso PIN permite ao usuário ter controle total sobre quando e quem pode ter acesso ao documento. Mas foi feita a escolha do código MRZ, priorizando a automação do processo e o maior tamanho da chave: 24 contra 4-6 de um PIN memorizável.

A recomendação oficial é de apenas ter cuidado no armazenamento e ocultamento do MRZ. Na prática, a segurança é semelhante aos documentos tradicionais. Um acesso não autorizado extrairia basicamente os mesmos dados impressos no MRZ (que o atacante já deve possuir). A criptografia adicional EAC deve proteger os demais dados mais sensíveis, como a biometria.

6.3 Procedimento em caso de perda ou roubo

Em caso de perda ou roubo do documento, a invalidação do número do documento é muito mais simples. No caso de uso em fraudes, há a detecção do documento roubado no ato da validação.

Para os MRTDs, casos de roubo ou perda de ePassports devem ser notificados a Interpol. Para o RIC, um documento roubado se torna inválido em todo o país.

6.4 Autenticação com chip defeituoso

Procedimento manual realizado com os documentos tradicionais. Tanto o MRTD como o RIC ainda possuem o mesmo valor como documento impresso. Portanto, ainda é possível tentativas de fraudes com os mesmos métodos atuais.

Apesar de possível, uma primeira falha na autenticação já chamaria a atenção de um inspetor. Ele também já estaria preparado para evitar justamente este tipo de fraude, tornando muito improvável o sucesso do criminoso.

É importante ressaltar que este tipo de autenticação é uma exceção. A permissão constante de acesso de documentos com chips defeituosos (ou a não utilização do sistema de inspeção) invalida todos os benefícios do MRTD.

6.5 Viabilidade técnica

O MRTD possui depende de muitos acordos de cooperação entre os países. Além dos repositórios de certificados não claramente definidos e incompletos, também não foi definido padrões de criptografia importantes como o EAC (seção 5.2). Já o RIC depende apenas de como serão implementados os requisitos opcionais.

Mesmo com o grande fluxo de passageiros ao redor do mundo, o maior investimento no caso do MRTD é na emissão e controle dos ePassports. Para o RIC, o alto investimento em terminais pode tornar a adoção limitada a poucos pontos, restringindo o potencial da tecnologia no Brasil.

Como o MRTD não foi planejado para o uso pretendido do RIC, toda a infraestrutura de CA, PKD e CRL seria subutilizado por se resumir a apenas um país. Por outro lado, o grande número de terminais (considerando que sejam de baixo custo) espalhados pelo Brasil podem criar uma demanda extremamente alta de consultas a lista de documentos inválidos, por exemplo.

Outro grande problema da proposta do MRTD é não levar em consideração questões diplomáticas. Os certificados dos países (CCSCA) devem ser distribuídos de forma estritamente segura e diplomática entre os países (seção 4.5.2). Há casos onde este meio simplesmente não existe (ex.: Irã e Israel).

7 Conclusões

Após a análise de todas os procedimentos de segurança do MRTD, é possível afirmar que ele foi elaborado focando muito mais a padronização no controle dos órgãos emissores de passaporte do que garantir o sigilo das informações do usuário. Por exemplo, o BAC não requer necessariamente a criptografia dos dados gravados no cartão.

Outra evidência da falta de cuidado com a privacidade, é que não há nenhuma especificação de como devem ser os sistemas de inspeção dos documentos, facilitando ainda mais o roubo de dados. É importante ressaltar porém, que a maioria destas informações são simples e já estão impressas no próprio documento (nome completo, data de nascimento, etc).

O mais preocupante é que em um modelo de documento com suporte a biometria, a proteção destes dados absolutamente confidenciais (impressões digitais e íris) não tenham sido padronizados pela ICAO. Falhas de segurança permitiriam não apenas acesso, mas também captação destas informações por sistemas de inspeção mal intencionados.

Neste sentido, a ICAO enfatiza que todo o processo de autenticação deve se acompanhado pessoalmente por um fiscal. O MRTD é uma ferramenta de fiscalização, e não deve substituir um humano. No cenário do RIC, esta recomendação é inviável e portanto é de extrema importância que neste projeto todos os mecanismos de segurança sejam implementados.

7.1 Funcionalidades adicionais

Neste estudo foi apresentado a estensão do RIC com o SGCA. A incorporação de certificados de atributos é uma forma segura de agregar informações ao RIC sem interferir nos processos de autenticação.

A tecnologia javacard permite que aplicações mais complexas sejam instaladas no cartão, porém com a vulnerabilidade do RFID e a necessidade de proteção do código MRZ, funcionalidades adicionais devem ser estudadas com muito cuidado para se manter o sigilo dos dados e não comprometer a segurança das outras aplicações.

Por exemplo, no SGCA deste estudo não foi implementado nenhuma forma de proteção de acesso a cada atributo. Assim, é possível extrair todos os tipos de atributo que um RIC possui. Um trabalho futuro seria o controle de nível de acesso de cada de atributo, limitando cada instituição a visualizar somente os atributos de seu interesse.

7.2 Considerações finais

Em um cenário onde todos os requisitos propostos pela ICAO sejam obedecidos, o ePassport é extremamente eficiente na automação e segurança. Mesmo com as ressalvas na falta de documentação sobre como devem ser os equipamentos de leitura dos cartões e repositório global de certificados.

A maioria das vulnerabilidades encontradas neste estudo é de simples correção (implementação de requisitos opcionais). Na prática, elas existem por causa da grande diversidade dos países envolvidos. Exceções manuais e a lenta adesão dos países inutiliza requisitos importantes. Por exemplo, o EAC é opcional e definido por cada país, mesmo sendo extremamente importante criptografar os dados biométricos.

No contexto do RIC, apenas as vantagens seriam incorporadas. Toda a complexidade da gestão de um serviço de PKD e CSCA se resume a apenas um país com total autonomia para definir como irá implementar cada um dos requisitos.

Na prática, o uso no cotidiano do RIC deverá ser o mesmo do documento impresso atual. A maior diferença será na drástica redução burocrática na emissão, facilidade de consulta e revogação de documentos, itens importantes para o governo no combate a fraudes.

A privacidade das informações contidas no RIC assim como a proteção contra fraudes dependerá muito em como serão implementados os terminais de consulta. Mas é provável que estes sistemas sejam utilizados apenas em situações especiais (órgãos públicos, contratos, etc) devido ao alto custo e a necessidade de homologação.

O ePassport está longe de se provar uma evolução do passaporte atual. Já o RIC possui o potencial de se tornar um modelo de aplicação do Doc 9303 justamente por poder atender a todos os requistos propostos.

Referências

CONVENçãO sobre Aviação Civil Internacional. Disponível em: <http://en.wikipedia.org/wiki/ChicagoConventiononInternationalCivilAviation>. Acesso em: 16 set. 2013.

DOCUMENT 9303: Document 9303. Disponível em: <http://www.icao.int/Security/mrtd/Pages/Document9303.aspx>. Acesso em: 22 mar. 2013.

EPASSPORTS Reloaded. Disponível em: <https://www.blackhat.com/presentations/bh-usa-08/vanBeek/\bhus08vanBeekePassportsReloadedSlides.pdf>. Acesso em: 16 set. 2013.

FINGERPRINTING Passports. Disponível em: <http://www.cs.ru.nl/~erikpoll/papers/nluug.pdf>. Acesso em: 22 ago. 2013.

ICAO: International civil aviation Organization. Disponível em: <http://www.icao.int/>. Acesso em: 18 fev. 2013.

ICP-Brasil. *DOC-ICP-16*. 2012. Disponível em: <http://www.iti.gov.br/component/content/article/143-icp-brasil/legislacao/790-doc-icp/>. Acesso em: 04 jun. 2013.

JAVA Card Technology. Disponível em: <http://www.oracle.com/technetwork/java/javacard/downloads/index.html>. Acesso em: 18 fev. 2013.

JMRTD. *Javacard Machine Readable Travel Document*. 2010. Disponível em: <http://jmrtd.org/>. Acesso em: 04 jun. 2013.

MRTD: Machine readable travel documents programme. Disponível em: <http://www.icao.int/Security/mrtd/{Pages}/default.aspx>. Acesso em: 18 fev. 2013.

RFC5755: An internet attribute certificate profile for authorization. Disponível em: <http://tools.ietf.org/html/rfc5755>. Acesso em: 22 ago. 2013.

RIC: Novo registro de identidade civil - portal brasil. Disponível em: <http://www.brasil.gov.br/para/servicos/documentacao/conheca-o-novo-registro-de-identidade-civil-ric>. Acesso em: 18 fev. 2013.

SGCA: Sistema de gerenciamento de certificados de atributos. Disponível em: <https://projetos.labsec.ufsc.br/sgca-iti/>. Acesso em: 04 jun. 2013.

STRESSING Security Highlights: Mrtd report vol.2 n.2. Disponível em: <http://www.icao.int/publications/journalsreports/2007/>. Acesso em: 05 set. 2013.

SUPLEMENT DOC 9303. Disponível em: <http://www.icao.int/Security/mrtd/Downloads/Supplements%20to\%20Doc%209303/Supplement%20to%20ICAO%20Doc%209303%20-%20Release12.pdf>. Acesso em: 16 set. 2013.

Anexos

ANEXO A – Passos na autenticação MRtd

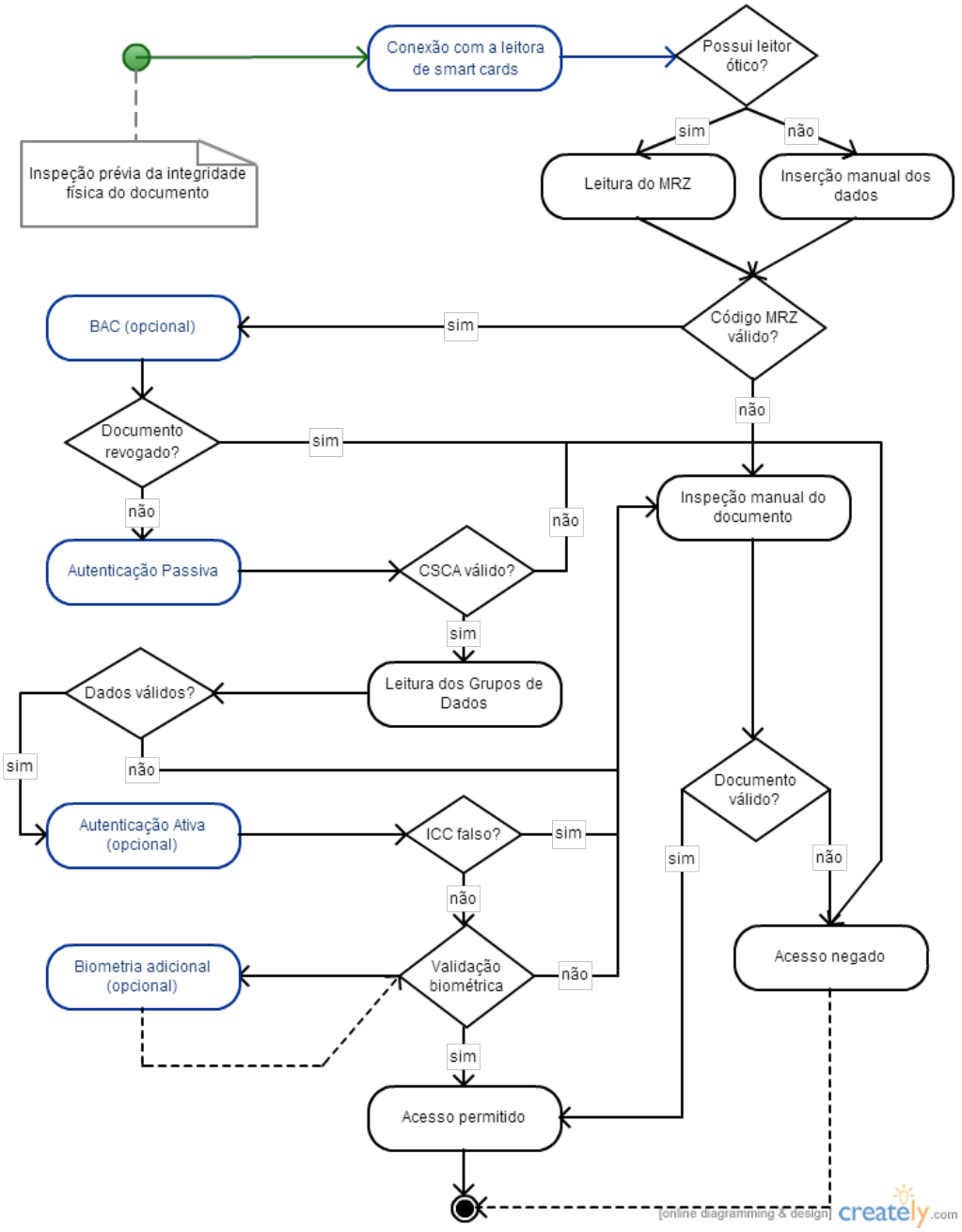

Figura 7 – Diagrama dos passos da autenticação de um MRtd

ANEXO B – Autenticação BAC do JMRTD v0.4.9

```
//class: PassportService extends PassportApduService
        /**
         * Inicializa o protocolo de Controle Básico de Acesso
         * @param bacKey é a chave extraída do código MRZ impresso no documento
         */
        public synchronized void doBAC(BACKeySpec bacKey){

                String documentNumber = bacKey.getDocumentNumber();
                String dateOfBirth    = bacKey.getDateOfBirth();
                String dateOfExpiry   = bacKey.getDateOfExpiry();

                /*Ocultado: validação das datas e número do documento*/

                //Extrai a Key Seed da chave BAC,
                //descrito na figura 6.2 da seção 6.1.3
                byte[] keySeed = Util.computeKeySeed(maxDocumentNumber, dateOfBirth, dateOfExpiry);

                SecretKey kEnc = Util.deriveKey(keySeed, Util.ENC_MODE);
                SecretKey kMac = Util.deriveKey(keySeed, Util.MAC_MODE);
                doBAC(kEnc, kMac);

        }

        /**
         * Continua a execução do protocolo BAC.
         * @param kEnc chave de encriptação 3DES
         * @param kMac chave de autentivação MAC
         */
        public synchronized void doBAC(SecretKey kEnc, SecretKey kMac){

                //Envia ao ICC o comando GET CHALLENGE, que gera
                //e retorna seu número aleatório
                byte[] rndICC = sendGetChallenge();

                //Gera os bytes aleatórios para o ICC e IFD
                byte[] rndIFD = new byte[8];
                random.nextBytes(rndIFD);
                byte[] kIFD = new byte[16];
                random.nextBytes(kIFD);

                //Envia o comando MUTUAL AUTHENTICATE para o ICC
                byte[] response = sendMutualAuth(rndIFD, rndICC, kIFD, kEnc, kMac);

                //Extrai a chave kICC da resposta
                byte[] kICC = new byte[16];
                System.arraycopy(response, 16, kICC, 0, 16);

                //Gera a Key seed de sessão com base em kIFD e kICC,
                //descrito na figura 6.2 da seção 6.1.3
                byte[] keySeed = new byte[16];
                for (int i = 0; i < 16; i++) {
```

```
                keySeed[i] = (byte) ((kIFD[i] & 0xFF) ^ (kICC[i] & 0xFF));
            }

            //Define as chaves de sessão para comunicação segura
            SecretKey ksEnc = Util.deriveKey(keySeed, Util.ENC_MODE);
            SecretKey ksMac = Util.deriveKey(keySeed, Util.MAC_MODE);

            //Cálculo do send sequence counter, também utilizado
            //na comunicação segura
            long ssc = Util.computeSendSequenceCounter(rndICC, rndIFD);
            wrapper = new SecureMessagingWrapper(ksEnc, ksMac, ssc);

            //Finaliza o protocolo
            BACEvent event = new BACEvent(this, rndICC, rndIFD, kICC, kIFD, true);
            notifyBACPerformed(event);
            state = BAC_AUTHENTICATED_STATE;
        }

    //class: PassportApduService extends CardService

        public synchronized byte[] sendMutualAuth(byte[] rndIFD, byte[] rndICC, byte[] kIFD, SecretKey kEn

            /*Ocultado: validação das variáveis recebidas*/

            //Encriptação com a chave kEnc
            cipher.init(Cipher.ENCRYPT_MODE, kEnc, ZERO_IV_PARAM_SPEC);

            //Concatena os dados para enviar ao ICC
            byte[] plaintext = new byte[32];
            System.arraycopy(rndIFD, 0, plaintext, 0, 8);
            System.arraycopy(rndICC, 0, plaintext, 8, 8);
            System.arraycopy(kIFD, 0, plaintext, 16, 16);
            byte[] ciphertext = cipher.doFinal(plaintext);

            //Gera o MAC dos dados
            mac.init(kMac);
            byte[] mactext = mac.doFinal(Util.pad(ciphertext));

            byte p1 = (byte) 0x00;
            byte p2 = (byte) 0x00;

            //Concatena os dados + MAC
            byte[] data = new byte[32 + 8];
            System.arraycopy(ciphertext, 0, data, 0, 32);
            System.arraycopy(mactext, 0, data, 32, 8);

            int le = 40; //tamanho total do dados a serem enviados

            //Envia o comando APDU ao ICC e aguarda a resposta
            CommandAPDU capdu = new CommandAPDU(ISO7816.CLA_ISO7816,ISO7816.INS_EXTERNAL_AUTHENTICATE,
            ResponseAPDU rapdu = transmit(capdu);
            byte[] rapduBytes = rapdu.getBytes();

            //Verifica o tamanho correto da resposta esperada
            if (rapduBytes.length != 42) {
                    throw new CardServiceException(
                            "Mutual␣authentication␣failed:␣expected␣length:␣40␣+␣2" +
                            rapduBytes.length, sw);
            }
```

```
        //O ICC deve extrair o rndICC da APDU e o compara
        //com o seu rndICC original. Se estiver correto,
        //é feita a concatenação S = rndICC+rndIFD+kICC
        //a encriptação com a chave kEnc: = E[kEnc](S)
        //e o MAC dos dados, e a resposta é montada de
        //modo análogo ao feito pelo IFD.

        //Decripta a resposta do ICC com o chave kEnc
        cipher.init(Cipher.DECRYPT_MODE, kEnc, ZERO_IV_PARAM_SPEC);
        byte[] result = cipher.doFinal(rapduBytes, 0, rapduBytes.length - 8 - 2);

        return result;

    }
```

ANEXO C – Applet SGCA

```
package labsec;
import javacard.framework.*;

public class SGCA extends Applet {

        //Codigo CLA do cabecalho APDU
        final static byte APPLET_CLA = (byte)0x80;

        //Certificado de Atributo
        private byte[] CA;
        //tamanho maximo do certificado
        private short CA_LENGTH = 128;

        //Codigos INS do cabecalho APDU
        final static byte GET = (byte) 0x10;
        final static byte SET = (byte) 0x20;

        //Codigo de falha: nenhum CA encontrado
        final static short SW_NO_CA = 0x6300;
        //Codigo de falha: tamanho max. do novo CA excedido
        final static short SW_CA_MAX_SIZE = 0x6301;

         private SGCA() {

                //Aloca toda a memoria necessaria pelo applet
                CA = new byte[CA_LENGTH];
                //Registra a instancia do applet no JCRE
                //e faz a referencia ao seu AID: 0102030405060708090000
                register();

         }

         /**
          * Installs this applet.
          * @param bArray the array containing installation parameters
          * @param bOffset the starting offset in bArray
          * @param bLength the length in bytes of the parameter data in bArray
          */
         public static void install (byte bArray[], short bOffset, byte bLength) throws ISOException

                //Para criar uma instancia do applet, o JCRE invoca este metodo primeiro.
                //Apos o applet inicializar suas variaveis, o metodo register() deve ser invocado.
                //Uma instalacao bem sucedida permite ao applet ser selecionado pela APDU SELECT.
                new SGCA();

         }

         /**
          * Processes an incoming APDU.
          * @see APDU
          * @param apdu the incoming APDU
          * @exception ISOException with the response bytes per ISO 7816-4
          */
         public void process(APDU apdu) throws ISOException {
```

```
        //Seleciona os bytes do cabecalho APDU: [CLA, INS, P1, P2]
        byte[] buffer = apdu.getBuffer();

        //Verifica se o comando INS nao e de selecao
        if (selectingApplet())
                return;

        //Verifica se o campo CLA do comando corresponde ao valor do applet
    if (buffer[ISO7816.OFFSET_CLA] != APPLET_CLA){
       ISOException.throwIt(ISO7816.SW_CLA_NOT_SUPPORTED);
    }

    //Realiza a chamada dos metodos correspondentes ao byte INS
    switch (buffer[ISO7816.OFFSET_INS]) {
                case GET:
                    getCA(apdu);
                    return;
                case SET:
                    setCA(apdu);
                    return;
                default:
                    ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
    }
 }

 private void getCA(APDU apdu) {

        //Verifica se existe um CA armazenado
        if(CA.length < 1){
                ISOException.throwIt(SW_NO_CA);
        }

        //Indica que esta APDU possui dados de entrada e saida
        apdu.setIncomingAndReceive();
        //Inicia a construcao da APDU de resposta
        apdu.setOutgoing();
        //Indica o tamanho em bytes da resposta
        apdu.setOutgoingLength((short)CA_LENGTH);
        //Envia os bytes do CA, apartir da posicao 0 do buffer
        apdu.sendBytesLong(CA, (short)0, (short)CA_LENGTH);

 }

private void setCA(APDU apdu) {


        //Seleciona os bytes do cabecalho APDU
        byte buffer[] = apdu.getBuffer();
        //Indica que esta APDU possui dados de entrada e saida
        short bytesRead = apdu.setIncomingAndReceive();

        //Cria uma variavel local do novo CA
        byte[] newCA = new byte[CA_LENGTH];
        //Ponto de inicio da leitura dos dados
        short echoOffset = (short)0;

        //Copia os dados recebidos no buffer para a variavel newCA
        while ( bytesRead > 0 ) {
```

```
                Util.arrayCopyNonAtomic(
                        buffer,                 //Fonte de Dados
                        ISO7816.OFFSET_CDATA,//Ponto de Inicio do buffer APDU
                        newCA,                  //Destino dos Dados
                        echoOffset,             //Ponto de Inicio
                        bytesRead);             //Quantidade a copiar

                        echoOffset += bytesRead;
                        bytesRead = apdu.receiveBytes(ISO7816.OFFSET_CDATA);
            }

            //Verifica se o tamanho do novo CA excede o limite
            if(newCA.length > CA_LENGTH){
                    ISOException.throwIt(SW_CA_MAX_SIZE);
            }

            //Atualiza o novo CA
            CA = newCA;

            //Inicia a construcao da APDU de resposta
            apdu.setOutgoing();
            //Indica o tamanho em bytes da resposta
            apdu.setOutgoingLength((short) (echoOffset));
            //Retorno com o novo CA gravado
            apdu.sendBytesLong(newCA, (short) 0, echoOffset);

        }

}
```

ANEXO D – Aplicação JavaAPDU

```
import java.util.List;

import javax.smartcardio.*;

public class JavaAPDU {


    public static void main(String[] args) throws CardException {

        System.out.println("Iniciando conexao...");

        TerminalFactory factory = TerminalFactory.getDefault();
        List<CardTerminal> terminals = factory.terminals().list();
        System.out.println("Leitoras encontradas: " + terminals);

        CardTerminal terminal = terminals.get(0);

        if(!terminals.isEmpty()){

            System.out.println("Conectado ao terminal: "+terminals);
            //Conecta ao primeiro terminal
            Card card = terminal.connect("T=1");
            System.out.println("Conectado ao cartao: " + card+"\n");
            //Verifica a disponibilidade do cartao
            card.getATR();

            //Seleciona o canal de comunicacao
            CardChannel channel = card.getBasicChannel();

            //Seleciona o Applet SGCA
            System.out.println("Selecionando Applet SGCA");
            byte[] aid = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x00, 0x00};
            ResponseAPDU answer = channel.transmit(new CommandAPDU(0x00, 0xA4, 0x04, 0x00, aid));
            System.out.println(answer.toString()+"\n");

            //Verifica o CA atual
            System.out.println("Certificado de Atributo atual:");
            byte[] cmd = {0x00, 0x00};
            ResponseAPDU get = channel.transmit(new CommandAPDU(0x80, 0x10, 0x00, 0x00, cmd));
            System.out.println(get.toString());
            //Converte os bytes lidos para hexadecimais
            byte[] convert = get.getBytes();
            System.out.println(printBytes(convert)+"\n");

            //Altera o Certificado de Atributo atual
            System.out.println("Novo Certificado de Atributo:");
            byte[] cmd2 = {0x0A, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x01, 0x02, 0x03, 0x04, 0x05};
            ResponseAPDU set = channel.transmit(new CommandAPDU(0x80, 0x20, 0x00, 0x00, cmd2));
            System.out.println(set.toString());
            byte[] update = set.getBytes();
            System.out.println(printBytes(update)+"\n");

            //Verifica o novo CA
            System.out.println("Certificado de Atributo atual:");
```

```
            ResponseAPDU get2 = channel.transmit(new CommandAPDU(0x80, 0x10, 0x00, 0x00, cmd));
            System.out.println(get2.toString());
            //Converte os butes para hexa
            byte[] answer2 = get2.getBytes();
            System.out.println(printBytes(answer2)+"\n");

            System.out.println("Encerrando conexoes...");
            card.disconnect(true);
            System.out.println("Desconectado");
        }
    }

    static public String printBytes(byte[] bytes){

        StringBuilder sb = new StringBuilder();
        for (byte b : bytes) {
            sb.append(String.format("%02X ", b));
        }

        return sb.toString();
    }

}
```

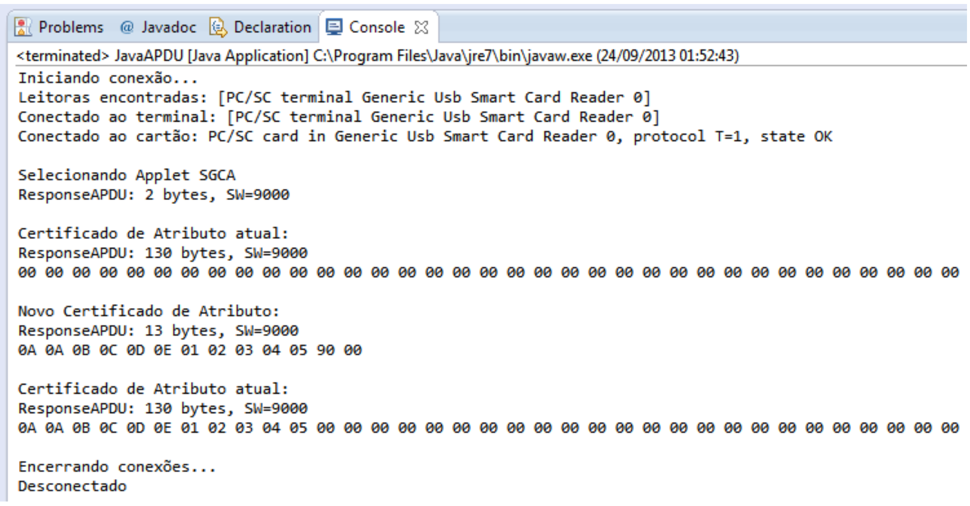

Figura 8 – Exemplo de execução da aplicação JavaAPDU